UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/667,122	09/21/2000	William C. Moyer	SC11306TH	9170

7590 01/30/2004

Motorola Inc
Austin Intellectual Property Law Section
MD TX32/PL02
7700 West Parmer Lane
Austin, TX 78729

EXAMINER	
HUISMAN, DAVID J	

ART UNIT	PAPER NUMBER
2183	

DATE MAILED: 01/30/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Offic Action Summary	Application No.	Applicant(s)	
	09/667,122	MOYER ET AL.	
	Examiner	Art Unit	
	David J. Huisman	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondenc address --

Peri d for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *08 January 2004*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-22* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-22* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. §§ 119 and 120

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

13)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application) since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.

a) ☐ The translation of the foreign language provisional application has been received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121 since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____.
4) ☐ Interview Summary (PTO-413) Paper No(s). _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: .

DETAILED ACTION

1. Claims 1-22 have been examined.

Papers Submitted

2. It is hereby acknowledged that the following papers have been received and placed of record in the file: Amendment "B" as received on 1/8/2004.

3. Claim 3 is objected to because of the following informalities: Please insert a semicolon after "memory" in line 3. Appropriate correction is required.

Withdrawn Rejections

4. Applicant's arguments directed toward the rejection of claims 10 and 18 have been fully considered and are found persuasive. Therefore, the rejections for claims 10 and 18 are hereby withdrawn by the examiner. However, upon further consideration, a new ground(s) of rejection is made below.

Maintained Rejections

5. Applicant has failed to overcome the rejections set forth for claims 1-9 and 13-17 from the office action mailed on July 24, 2003. Consequently, these rejections are respectfully maintained by the examiner and copied below for applicant's convenience.

Maintained Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

7. Claims 1-9 and 13-17 are rejected under 35 U.S.C. 103(a) as being unpatentable over Samra, U.S. Patent No. 6,275,926, in view of Oh, U.S. Patent No. 5,594,765.

8. Referring to claim 1, Samra has taught a processing system for accessing memory, comprising:

a) an address bus for providing a current address and a previous address to memory. See Fig.1, component 165. Note that it is inherent that this memory bus will comprise an address bus since an address must be applied to memory in order to retrieve or store data within the memory.

b) a data bus for receiving information from memory. See Fig.1, component 165 and again note that it is inherent that this memory bus will comprise a data bus in order to transfer data between the CPU and the memory.

c) Samra has not taught generating a first sequence signal that when negated indicates that the current address may not be sequential to the previous address. However, Oh has taught such a signal in a counter system used in conjunction with an SDRAM, which is a type of memory that can be found in Samra's system (see column 4, lines 30-35). A person of ordinary skill in the art would have recognized that Oh's system would allow Samra to incorporate burst transfers between the CPU and memory through use of a single address, thereby reducing activity on the address bus since all subsequent addresses are generated from the original address. This

automatic address generation also results in higher speeds since the CPU does not have to generate a new address for each memory access. As a result it would have been obvious to one of ordinary skill in the art at the time of the invention to combine Samra's system with that of Oh in order to allow Samra to take advantage of burst transfers. Regarding the first signal, Oh's system includes a signal "LATCH," which denotes the beginning of a burst cycle. See column 3, lines 38-39 and Fig.7. This signal indicates that the current address (being some address generated during burst mode) may not be sequential to the previous address because the burst mode may be set such that an access is occurring in non-sequential (interleaved) mode.

d) Samra has not taught generating a second sequence signal that when negated indicates that the current address is not sequential to the previous address. However, Oh has taught a second signal "seq_int#," which specifies either sequential or non-sequential (interleaved) burst mode. See column 1, lines 11-18, and column 3, lines 16-21. This signal, when negated will indicate that interleaved (or non-sequential mode) has been selected. In this situation, the current address will be non-sequential with respect to the previous address.

e) Finally, Samra has not explicitly taught generating a third sequence signal that when negated indicates that the current address, if it is an instruction address, is not sequential to the previous address that was an instruction address. However, Samra's system does perform branch prediction as shown in column 3, lines 1-8. As is known in the art of branch prediction, if a branch is predicted not-taken, then the current address (address of the instruction to be executed after the branch) will be sequential to the previous address (since the instruction following the branch, in program order, will be speculatively fetched based on the prediction). However, if it is determined that the branch direction was mispredicted, then a third signal must inherently exist

which indicates that a misprediction has occurred. Such a signal may cause a pipeline flush and the fetching of the appropriate instructions, for instance. For the example where the branch was predicted not-taken, this third signal would indicate that the current instruction address is not sequential to the previous instruction address. Instead, the current address would be the branch target address.

9. Referring to claim 2, Samra in view of Oh has taught a processing system as described in claim 1. Oh has further taught that if the current address is not sequential to the previous address, the first sequence signal is negated prior to the second sequence signal being negated. It should be noted that in order to be in sequential or interleaved burst mode, the system must first be in a general burst mode. Therefore, the first signal (LATCH) will provide an indication before the second signal (seq_int#).

10. Referring to claim 3, Samra has taught a processing system for accessing memory, comprising:

a) an address bus for providing a current address and a previous address to memory. See Fig. 1, component 165. Note that it is inherent that this memory bus will comprise an address bus since an address must be applied to memory in order to retrieve or store data within the memory.

b) a data bus for receiving information from memory. See Fig. 1, component 165 and again note that it is inherent that this memory bus will comprise a data bus in order to transfer data between the CPU and the memory.

c) an execution unit which generates branch conditions and data addresses. See Fig. 2, component 255, and column 5, lines 1-6. Note that the execution unit 255 produces flag results

(conditions based on operations) and also note that data is fetched by the execution unit 255 from the level-1 data cache, which means that it must generate data addresses.

d) a decode control unit which decodes instructions. See Fig.2, component 230.

e) Samra has not taught a fetch unit, coupled to the execution unit, the decode control unit, the address bus, and the data bus, for generating a first sequence signal that when negated indicates that the current address may not be sequential to the previous address. However, Oh has taught such a signal in a counter system used in conjunction with an SDRAM, which is a type of memory that can be found in Samra's system (see column 4, lines 30-35). A person of ordinary skill in the art would have recognized that Oh's system would allow Samra to incorporate burst transfers between the CPU and memory through use of a single address, thereby reducing activity on the address bus since all subsequent addresses are generated from the original address. This automatic address generation also results in higher speeds since the CPU does not have to generate a new address for each memory access. As a result it would have been obvious to one of ordinary skill in the art at the time of the invention to combine Samra's system with that of Oh in order to allow Samra to take advantage of burst transfers. Regarding the first signal, Oh's system includes a signal "LATCH," which denotes the beginning of a burst cycle. See column 3, lines 38-39 and Fig.7. This signal indicates that the current address (being some address generated during burst mode) may not be sequential to the previous address because the burst mode may be set such that an access is occurring in non-sequential (interleaved) mode.

f) Samra has not taught a unit for generating a second sequence signal that when negated indicates that the current address is not sequential to the previous address. However, Oh has taught a second signal "seq_int#," which specifies either sequential or non-sequential

(interleaved) burst mode. See column 1, lines 11-18, and column 3, lines 16-21. This signal, when negated will indicate that interleaved (or non-sequential mode) has been selected. In this situation, the current address will be non-sequential with respect to the previous address.

g) Samra has not explicitly taught a unit for generating a third sequence signal that when negated indicates that the current address, if it is an instruction address, is not sequential to the previous address that was an instruction address. However, Samra's system does perform branch prediction as shown in column 3, lines 1-8. As is known in the art of branch prediction, if a branch is predicted not-taken, then the current address (address of the instruction to be executed after the branch) will be sequential to the previous address (since the instruction following the branch, in program order, will be speculatively fetched based on the prediction). However, if it is determined that the branch direction was mispredicted, then a third signal must inherently exist which indicates that a misprediction has occurred. Such a signal may cause a pipeline flush and the fetching of the appropriate instructions, for instance. For the example where the branch was predicted not-taken, this third signal would indicate that the current instruction address is not sequential to the previous instruction address. Instead, the current address would be the branch target address.

11. Referring to claim 4, Samra in view of Oh has taught a processing system as described in claim 3. Samra in view of Oh has not explicitly taught that the decode control unit comprises an instruction register. However, Official Notice is taken that an instruction register and its purpose is well known and expected in the art. The instruction register is used to store an instruction which will be decoded into signals used to control the operation of the processor. Consequently,

it would have been obvious to one of ordinary skill in the art at the time of the invention to provide such a component so that a particular instruction can be decoded.

12. Referring to claim 5, Samra in view of Oh has taught a processing system as described in claim 3. Samra in view of Oh has further taught that the fetch unit comprises an address control unit, coupled to the decode control unit (Fig.2, component 230) and the execution unit (Fig.2, component 255), for receiving a branch condition signal from the execution unit (column 5, lines 1-8 and note in Fig.2 that flag results are received) and a branch decode signal and a load/store signal (it is inherent that if a branch is to be executed then a branch decode signal will be provided. Likewise, if a load/store is to be executed, then the appropriate signal should be provided) from the decode unit and for providing the first, second, and third sequence signal. It should be noted by the applicant that the address control unit could be broadly interpreted as comprising the branch unit 270, the load store unit 265, the BTB 275, the fetcher 215 (all components taught by Samra in Fig.2), and the component taught by Oh, allowing for burst transfers. These components all play a part in determining what addresses are applied to memory and therefore, they form a unit which controls addressing. This unit also would be responsible for providing all three of the sequence signals (the first two from Oh's system and the third from the branch's execution).

13. Referring to claim 6, Samra in view of Oh has taught a processing system as described in claim 5. Samra has further taught that the execution unit comprises a condition generator that provides the branch condition signal. See column 5, lines 1-8 and note that the execution unit provides flags (conditions) which are used to determine whether or not a branch is taken.

14. Referring to claim 7, Samra in view of Oh has taught a processing system as described in claim 6. Samra has further taught that the execution unit comprises a data address generator which provides a data address signal to the fetch unit. Note from Fig.2 that data is fetched by the execution unit 255 from the level-1 data cache, which means that it must provide data addresses to the logic that utilizes the generated address in order to fetch from the cache.

15. Referring to claims 8 and 9, Samra in view of Oh has taught a processing system as described in claims 7 and 3, respectively. Oh has further taught that if the current address is not sequential to the previous address, the first sequence signal is negated prior to the second sequence signal being negated. It should be noted that in order to be in sequential or interleaved burst mode, the system must first be in a general burst mode. Therefore, the first signal (LATCH) will provide an indication before the second signal (seq_int#).

16. Referring to claim 13, Samra has taught a processing system comprising:

a) an execution unit. See Fig.2, components 255, 260, 265, and 270.

b) a decode control unit. See Fig.2, component 230.

c) Samra has not taught a fetch unit, coupled to the execution unit and the decode control unit, for providing addresses on an address bus which may be sequential and providing a first sequence signal which indicates if a current address may be sequential to a previous address. However, Oh has taught such a signal in a counter system used in conjunction with an SDRAM, which is a type of memory that can be found in Samra's system (see column 4, lines 30-35). A person of ordinary skill in the art would have recognized that Oh's system would allow Samra to incorporate burst transfers between the CPU and memory through use of a single address, thereby reducing activity on the address bus since all subsequent addresses are generated from

the original address. This automatic address generation also results in higher speeds since the CPU does not have to generate a new address for each memory access. As a result it would have been obvious to one of ordinary skill in the art at the time of the invention to combine Samra's system with that of Oh in order to allow Samra to take advantage of burst transfers. Regarding the first signal, Oh's system includes a signal "LATCH," which denotes the beginning of a burst cycle. See column 3, lines 38-39 and Fig.7. This signal indicates that the current address (being some address generated during burst mode) may be sequential to the previous address because the burst mode may be set such that an access is occurring in sequential mode.

d) Samra has not taught a unit for providing a second sequence signal which indicates if the current address is sequential to the previous address. However, Oh has taught a second signal "seq_int#," which specifies either sequential or non-sequential (interleaved) burst mode. See column 1, lines 11-18, and column 3, lines 16-21. This signal, when asserted, will indicate that sequential mode has been selected. In this situation, the current address will be sequential with respect to the previous address.

17. Referring to claim 14, Samra in view of Oh has taught a processing system as described in claim 13. Oh has further taught that if the second sequence signal indicates that the current address is not sequential to the previous address, the first sequence signal indicates that the current address may not be sequential to the previous address prior to the second sequence signal indicating that the current address is not sequential to the previous address.. It should be noted that in order to be in sequential or interleaved burst mode, the system must first be in a general burst mode. Therefore, the first signal (LATCH) will provide an indication before the second signal (seq_int#).

18. Referring to claim 15, Samra in view of Oh has taught a processing system as described in claim 14. Samra has not explicitly taught generating a third sequence signal which indicates if a current instruction address is sequential to a previous instruction address. However, Samra's system does perform branch prediction as shown in column 3, lines 1-8. As is known in the art of branch prediction, if a branch is predicted not-taken, then the current address (address of the instruction to be executed after the branch) will be sequential to the previous address (since the instruction following the branch, in program order, will be speculatively fetched based on the prediction). However, if it is determined that the branch direction was mispredicted, then a third signal must inherently exist which indicates that a misprediction has occurred. Such a signal may cause a pipeline flush and the fetching of the appropriate instructions, for instance. For the example where the branch was predicted not-taken, this third signal would indicate that the current instruction address is not sequential to the previous instruction address. Instead, the current address would be the branch target address.

19. Referring to claim 16, Samra in view of Oh has taught a processing system as described in claim 15. Samra has further taught that the execution unit comprises a condition generator that provides a branch condition signal to the fetch unit. See column 5, lines 1-8 and note that the execution unit 255 provides flags (conditions) which are used to determine whether or not a branch is taken.

20. Referring to claim 17, Samra in view of Oh has taught a processing system as described in claim 16. Samra has further taught that the decode control unit provides a branch decode signal and a load/store signal to the fetch unit. For instance, see Fig.2 and note that the branch unit 270 and load-store unit 265 receive decoded information sent along by the decoder 230.

Furthermore, it is inherent that if a branch is to be executed then a branch decode signal will be provided. Likewise, if a load/store is to be executed, then the appropriate signal should be provided. These signals direct the operation of the appropriate execution units.

New Claim Rejections - 35 USC § 102

21. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

22. Claims 10 and 18-20 are rejected under 35 U.S.C. 102(b) as being anticipated by Kromer, III, U.S. Patent No. 4,541,045 (herein referred to as Kromer).

23. Referring to claim 10, Kromer has taught a processing system for fetching instructions and data, comprising:

a) an address bus for providing a current address for retrieving a first instruction, a previous address for retrieving a second instruction, and a data address for retrieving data, wherein the data address occurs before the current address and after the previous address. See Fig.1, address bus 11, and also Fig.2A. Note from Fig.2A, that a current address 3 is provided, a previous address 2 is provided, and a data address AD_0 occurs before the current address and after the previous address.

b) a data bus for retrieving the first and second instructions and the data. See Fig.1, and notice instruction/data bus 13. From Fig.2A, it can be seen that this bus receives instructions and data (I_0, I_1, D_0, D_1...).

c) a fetch unit, coupled to the address bus and the data bus, for generating a first sequence signal that when asserted for the current address indicates that the current address is sequential to the previous address and when negated indicates that the current address may not be sequential to the previous address. From Fig.1, it should be noted that all components within the system are interconnected, which include the fetch unit (which mainly comprises components 11, 15, 17, and 19). In addition, see Fig.2A and note that instruction addresses (0, 1, 2...81, 82, 83) and data addresses (AD_0, AD_1...AD_5) are provided on the address bus. As can be seen from the addresses on the address bus, the instruction addresses are sequential until a branch is encountered. For as long as the instruction addresses are sequential, signal VALID +.5 (Fig.2C) remains high. However, a branch instruction, when taken or unconditional, inherently changes program flow. For instance, looking at Fig.3, the branch (jump) instruction (which comprises instructions 3 and 4) causes the program to jump to address 80, as opposed to sequentially fetching the next instruction. This jump is also illustrated in Fig.2A (notice how address 80 follows address 5). From column 7, line 58, to column 8, line 18, it is disclosed that the VALID +.5 signal goes low when the branch destination (80) is encountered, in this case for instruction I_4. This is seen in Fig.2C. Therefore, when VALID +.5 is negated, this signal indicates that a branch has been encountered and the current instruction address is may not be sequential to a previous instruction address (as shown in Fig.2A and Fig.2C).

24. Referring to claim 18, Kromer has taught a processing unit comprising:

a) an execution unit. See Fig.1, component 41. In addition, it should be realized that most, if not all, of the components shown in Fig.1 can be considered an execution unit, since each component works together in order to execute instructions.

b) a decode control unit. See Fig.1, components 29 and 37.

c) a fetch unit, coupled to the execution unit and the decode unit, for providing instruction and data addresses on an address bus and providing a first sequence signal that indicates if a current instruction address is sequential to a previous instruction address even if a data address is provided between the current instruction address and the previous instruction address. From Fig.1, it should be noted that all components within the system are interconnected, which include the fetch unit (which mainly comprises components 11, 15, 17, and 19). In addition, see Fig.2A and note that instruction addresses (0, 1, 2...81, 82, 83) and data addresses (AD_0, AD_1...AD_5) are provided on the address bus. As can be seen from the addresses on the address bus, the instruction addresses are sequential until a branch is encountered. A branch, when taken or unconditional, inherently changes program flow. For instance, looking at Fig.3, the branch (jump) instruction (which comprises instructions 3 and 4) causes the program to jump to address 80, as opposed to sequentially fetching the next instruction. This jump is also illustrated in Fig.2A (notice how address 80 follows address 5). From column 7, line 58, to column 8, line 18, it is disclosed that the VALID +.5 signal goes low when the branch destination (80) is encountered, in this case for instruction I_4. This is seen in Fig.2C. Therefore, when VALID +.5 is high, this signal indicates that a branch has not been encountered and the current instruction address is sequential to a previous instruction address (as shown in Fig.2A and Fig.2C). And, this holds true even if a data address is provided between the current instruction address and the previous instruction address, because from Fig.2A, sequential instruction addresses are provided even with data addresses in between them. For instance, between sequential instruction

addresses 2 and 3, data address AD_0 is provided. And, recall that the first sequence signal (VALID +.5) being at a high level denotes sequential instruction addressing.

25. Referring to claim 19, Kromer has taught a processing unit as described in claim 18. Kromer has further taught that the execution unit comprises a condition generator that provides a branch condition signal to the fetch unit. Again, recall the VALID +.5 signal shown in Fig.2C. As previously discussed, this signal goes low when a branch instruction is encountered. Therefore, when a branch is encountered, a condition generator will generate the appropriate VALID +.5 signal (this is inherent because the signal must be generated by something). This signal is then used by the fetch unit to fetch an instruction at a non-sequential address. And, this signal is considered to be a branch condition signal because it makes components aware of the state (condition or mode) of the system, i.e., branch mode.

26. Referring to claim 20, Kromer has taught a processing unit as described in claim 19. Kromer has further taught that the decode control unit provides a branch decode signal and a load/store signal to the fetch unit. See Fig.1, signal 30, which is a decode signal (it comes from the decoder). This signal is used to control the program counter in branching (jump) situations (column 2, lines 62-63). In addition, a load/store signal is also provided by the same decoder. See the WRITE signal 60 in Fig.1. In column 3, lines 21-27, Kromer has taught that this signal is used to indicate a store.

27. Claims 13-14 and 22 are rejected under 35 U.S.C. 102(b) as being anticipated by Hennessy and Patterson, Computer Architecture - A Quantitative Approach, 2nd Edition, 1996 (herein referred to as Hennessy).

28. Referring to claim 13, Hennessy has taught a processing system comprising:

a) an execution unit. Note that the pipeline illustrated in Figure 3.2 on page 132 has an execution stage (EX). In order to perform execution, an execution unit must inherently exist.

b) a decode control unit. Note that the pipeline on page 132 has an instruction decode stage (ID). In order to perform decoding, a decode control unit must inherently exist.

c) a fetch unit, coupled to the execution unit and the decode control unit, for providing addresses on an address bus which may be sequential and providing a first sequence signal which indicates if a current address may be sequential to a previous address and a second sequence signal which indicates if the current address is sequential to the previous address. See page 163 and note that the fetch unit (in the fetch stage of the pipeline – IF) is connected to components in the other stages (ID and EX). In addition, Hennessy has disclosed the basic concept of branch prediction on pages 262-264 and 275. In essence, when a branch instruction is encountered (previous instruction address), a prediction is made as to whether the branch will be taken (next instruction address is not sequential to the branch instruction address) or not taken (next instruction address is sequential to the branch instruction address). This branch prediction is the first sequence signal in that it indicates if a current address may be sequential to a previous address. For example, if a branch is predicted not taken, then the current address may be sequential to the previous address. It may be sequential because this is merely a prediction and the actual address is not known yet. However, once it is officially determined whether the branch will be taken or not, the second sequence signal will be given (that is, the signal which represents the actual outcome of the branch). If the branch is not taken, then the signal will indicate that the current

address is sequential to the previous address. This signal would be used to verify the prediction. If the prediction was incorrect, then corrective measures would need to be taken.

29. Referring to claim 14, Hennessy has taught a processing system as described in claim 13. Hennessy has further taught that if the second sequence signal indicates that the current address is not sequential to the previous address, the first sequence signal indicates that the current address may not be sequential to the previous address prior to the second sequence signal indicating that the current address is not sequential to the previous address. Looking at pages 262-264 and 275 of Hennessy, it should be realized that if the branch is actually taken, the second sequence signal (actual outcome signal) will indicate that the current address is not sequential to the previous address. If the branch were predicted taken, then the first sequence signal (prediction signal) would indicate that the current address may not be sequential to the previous address. And, it should be realized that the first sequence signal will indicate before the second signal indicates since a branch prediction precedes a branch actual outcome.

30. Referring to claim 22, Hennessy has taught a processing unit as described in claim 14. Furthermore, it is inherent that the second sequence signal, which represents the actual branch outcome, is provided in response to resolving a branch condition code. This is because a conditional branch's outcome, which is the type of branch that is predicted, is dependent on resolving a condition code.

New Claim Rejections - 35 USC § 103

31. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

32. Claims 1-9 and 21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hennessy, as applied above, in view of Kromer, as applied above.

33. Referring to claim 1, Hennessy has taught a processing system for accessing memory, comprising:

a) an address bus for providing a current address and a previous address to memory. It is inherent that such a bus exists within a system. In general, when executing a program, address after address is presented on an address bus (this would include current and previous addresses) in order to fetch instructions and/or data.

b) a data bus for receiving information from memory. This is another inherent component. A data bus must exist such that data can be transferred to and from memory in response to an address on the address bus.

c) generating a first sequence signal that when negated indicates that the current address may not be sequential to the previous address. Hennessy has disclosed the basic concept of branch prediction on pages 262-264 and 275. In essence, when a branch instruction is encountered (previous instruction address), a prediction is made as to whether the branch will be taken (next instruction address is not sequential to the branch instruction address) or not taken (next instruction address is sequential to the branch instruction address). This branch prediction is the first sequence signal in that it indicates if a current address may be sequential to a previous address. For example, if a branch is predicted taken, then the current address may not be

sequential to the previous address. It may not be sequential because this is merely a prediction and the actual address is not known yet.

d) generating a second sequence signal that when negated indicates that the current address is not sequential to the previous address. Once it is officially determined whether the branch will be taken or not, the second sequence signal will be given (that is, the signal which represents the actual outcome of the branch). If the branch is taken, then the signal will indicate that the current address is not sequential to the previous address. This signal would be used to verify the prediction. If the prediction was incorrect, then corrective measures would need to be taken.

e) Hennessy has not taught generating a third sequence signal that when negated indicates that the current address, if it is an instruction address, is not sequential to the previous address that was an instruction address. However, Kromer has taught such a signal. From Fig.2A of Kromer, note that instruction addresses (0, 1, 2...81, 82, 83) are provided on the address bus. As can be seen from the addresses on the address bus, the instruction addresses are sequential until a branch is encountered. A branch, when taken or unconditional, inherently changes program flow. For instance, looking at Fig.3, the unconditional branch (jump) instruction (which comprises instructions 3 and 4) causes the program to jump to address 80, as opposed to sequentially fetching the next instruction. This jump is also illustrated in Fig.2A (notice how address 80 follows address 5). From column 7, line 58, to column 8, line 18, it is disclosed that the VALID +.5 signal goes low when the branch destination (80) is encountered, in this case for instruction I_4. This is seen in Fig.2C. Therefore, when VALID +.5 is high, this signal indicates that a branch has not been encountered and the current instruction address is sequential to a previous instruction address (as shown in Fig.2A and Fig.2C). Such a signal helps synchronize operation

flow so as to optimally utilize microprocessor cycles. See column 1, lines 58-61. And, although the first two signals in Hennessy deal with conditional branches, this signal would help optimization for unconditional branches (which don't rely on a condition, as the one shown in Fig.3 of Kromer). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Hennessy to include the third sequence signal (VALID +.5) as taught by Kromer, in order to increase optimization.

34. Referring to claim 2, Hennessy in view of Kromer has taught a processing unit as described in claim 1. Hennessy has further taught that if the current address is not sequential to the previous address, the first sequence signal is negated prior to the second sequence signal being negated. Looking at pages 262-264 and 275 of Hennessy, it should be realized that if the branch is actually taken, the second sequence signal (actual outcome signal) will indicate that the current address is not sequential to the previous address. If the branch were predicted taken, then the first sequence signal (prediction signal) would indicate that the current address may not be sequential to the previous address. And, it should be realized that the first sequence signal will indicate before the second signal indicates since a branch prediction precedes a branch actual outcome.

35. Referring to claim 3, Hennessy has taught a processing system for accessing memory, comprising:

a) an address bus for providing a current address and a previous address to memory. It is inherent that such a bus exists within a system. In general, when executing a program, address after address is presented on an address bus (this would include current and previous addresses) in order to fetch instructions and/or data.

b) a data bus for receiving information from memory. This is another inherent component. A data bus must exist such that data can be transferred to and from memory in response to an address on the address bus.

c) an execution unit which generates branch conditions and data addresses. Looking at page 134 of Hennessy, the execution stage comprises each of the components between the ID/EX and EX/MEM registers (the ALU, MUXs, zero-tester). As seen from the figure, the ALU is able to output an address which then goes to the data memory. In addition, the zero-tester generates a branch condition, i.e., in this case, it checks to see whether a value is zero or not (the condition being whether a value is zero).

d) a decode control unit which decodes instructions. Note on page 132 that a pipeline comprises an ID stage, which stands for "instruction decode." And, in order for instruction decoding to occur, a decode control unit must exist.

e) a fetch unit, coupled to the execution unit, the decode control unit, the address bus, and the data bus (note that the entire system on page 134, including those components in the fetch stage-IF, are coupled together) for generating a first sequence signal that when negated indicates that the current address may not be sequential to the previous address. Hennessy has disclosed the basic concept of branch prediction on pages 262-264 and 275. In essence, when a branch instruction is encountered (previous instruction address), a prediction is made as to whether the branch will be taken (next instruction address is not sequential to the branch instruction address) or not taken (next instruction address is sequential to the branch instruction address). This branch prediction is the first sequence signal in that it indicates if a current address may be sequential to a previous address. For example, if a branch is predicted taken, then the current

address may not be sequential to the previous address. It may not be sequential because this is merely a prediction and the actual address is not known yet.

f) Samra has not taught a unit for generating a second sequence signal that when negated indicates that the current address is not sequential to the previous address. Once it is officially determined whether the branch will be taken or not, the second sequence signal will be given (that is, the signal which represents the actual outcome of the branch). If the branch is taken, then the signal will indicate that the current address is not sequential to the previous address. This signal would be used to verify the prediction. If the prediction was incorrect, then corrective measures would need to be taken.

g) Hennessy has not explicitly taught a unit for generating a third sequence signal that when negated indicates that the current address, if it is an instruction address, is not sequential to the previous address that was an instruction address. Hennessy has not taught generating a third sequence signal that when negated indicates that the current address, if it is an instruction address, is not sequential to the previous address that was an instruction address. However, Kromer has taught such a signal. From Fig.2A of Kromer, note that instruction addresses (0, 1, 2...81, 82, 83) are provided on the address bus. As can be seen from the addresses on the address bus, the instruction addresses are sequential until a branch is encountered. A branch, when taken or unconditional, inherently changes program flow. For instance, looking at Fig.3, the unconditional branch (jump) instruction (which comprises instructions 3 and 4) causes the program to jump to address 80, as opposed to sequentially fetching the next instruction. This jump is also illustrated in Fig.2A (notice how address 80 follows address 5). From column 7, line 58, to column 8, line 18, it is disclosed that the VALID +.5 signal goes low when the branch

destination (80) is encountered, in this case for instruction I_4. This is seen in Fig.2C. Therefore, when VALID +.5 is high, this signal indicates that a branch has not been encountered and the current instruction address is sequential to a previous instruction address (as shown in Fig.2A and Fig.2C). Such a signal helps synchronize operation flow so as to optimally utilize microprocessor cycles. See column 1, lines 58-61. And, although the first two signals in Hennessy deal with conditional branches, this signal would help optimization for unconditional branches (which don't rely on a condition, as the one shown in Fig.3 of Kromer). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Hennessy to include the third sequence signal (VALID +.5) as taught by Kromer, in order to increase optimization.

36. Referring to claim 4, Hennessy in view of Kromer has taught a processing system as described in claim 3. Furthermore, Hennessy has taught that the decode control unit comprises an instruction register. See page 127 and note that the instruction is placed in the instruction register (IR) when it is fetched. The IR is accessed by the decode control unit in order to decode the instruction by fetching operands, extracting immediate values, etc.

37. Referring to claim 5, Hennessy in view of Kromer has taught a processing system as described in claim 4. Furthermore, Hennessy has taught an address control unit, coupled to the decode unit and the execution unit, for receiving a branch condition signal (note that the branch condition signal is used to control a MUX in the fetch stage) and a branch decode signal (it is inherent that a branch decode signal is produced so that appropriate instruction fetching occurs) and a load/store signal (again this is inherent because the system must know when it will perform

a load or store) and for providing the first, second, and third sequence signals (from the rejection of claim 3 above, it has been explained that Hennessy in view of Kromer produce such signals).

38. Referring to claim 6, Hennessy in view of Kromer has taught a processing system as described in claim 5. Hennessy has further taught that the execution unit comprises a condition generator that provides the branch condition signal. See page 134, and note the zero-tester is a condition generator. It generates a condition (whether data is 0 or not) and provides the appropriate branching signal which is sent back to the fetch stage.

39. Referring to claim 7, Hennessy in view of Kromer has taught a processing system as described in claim 5. Hennessy has further taught that the execution unit comprises a data address generator which provides a data address signal to the fetch unit. Note on page 134 that the output of the ALU is sent back to the fetch stage as an input into the MUX. This output is an address since it is used to address memory in the MEM (memory) stage.

40. Referring to claims 8 and 9, Hennessy in view of Kromer has taught a processing system as described in claims 7 and 3, respectively. Furthermore, Hennessy has taught that if the current address is not sequential to the previous address, the first sequence signal is negated prior to the second sequence signal being negated. Looking at pages 262-264 and 275 of Hennessy, it should be realized that if the branch is actually taken, the second sequence signal (actual outcome signal) will indicate that the current address is not sequential to the previous address. If the branch were predicted taken, then the first sequence signal (prediction signal) would indicate that the current address may not be sequential to the previous address. And, it should be realized that the first sequence signal will indicate before the second signal indicates since a branch prediction precedes a branch actual outcome.

41. Referring to claim 21, Hennessy in view of Kromer has taught a processing unit as described in claim 2. Furthermore, it is inherent that the second sequence signal, which represents the actual branch outcome, is negated in response to resolving a branch condition code. This is because a conditional branch's outcome, which is the type of branch that is predicted, is dependent on resolving a condition code.

42. Claims 11-12 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kromer, as applied above.

43. Referring to claim 11, Kromer has taught a processing system as described in claim 10. Kromer has further taught:

a) an address control unit for receiving a branch decode signal. See Fig.1, signal 30, which is a decode signal (it comes from the decoder). This signal is used to control the program counter in branching (jump) situations (column 2, lines 62-63).

b) the unit receives a load/store signal. See the WRITE signal 60 in Fig.1. In column 3, lines 21-27, Kromer has taught that this signal is used to indicate a store. In addition, the unit will provide the first sequence signal VALID +.5 as shown in Fig.2C.

c) Kromer has not explicitly taught that the unit receives a branch condition. However, Official Notice is taken that conditional branches and their advantages are well known, accepted, and expected in the art. These branches allow for more flexibility in that the program has the ability to branch based on whether a particular condition is met. For instance, you may want to perform a first task if an addition result equals 0 and a second task if an addition result is not zero. Conditional branches allow for such flexibility. And, it should be realized that conditional

branches are dependent on conditions. Therefore, with conditional branches, a branch condition must inherently be received in order to determine if the branch is taken or not taken. As a result, in order to achieve the increased flexibility of conditional branching, it would have been obvious to one of ordinary skill in the art at the time of the invention to implement conditional branches , whereby branch conditions are received.

44. Referring to claim 12, Kromer has taught a processing system as described in claim 11.

a) Kromer has further taught a decode control unit which provides the branch decode signal and the load/store signal. See Fig.1, signal 30, which is a decode signal (it comes from the decoder). This signal is used to control the program counter in branching (jump) situations (column 2, lines 62-63). In addition, a load/store signal is also provided by the same decoder. See the WRITE signal 60 in Fig.1. In column 3, lines 21-27, Kromer has taught that this signal is used to indicate a store.

b) Kromer has not explicitly taught an execution unit which provides the branch condition. However, Official Notice is taken that execution units provide branch conditions. A well known branch instruction is of the type "beq," which corresponds to "branch if the result is zero." To determine if the result is 0, the execution unit will provide the result (i.e., the branch condition). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to have an execution unit that provides a branch condition.

Response to Arguments

45. Applicant's arguments filed on January 8, 2004, have been fully considered but they are not persuasive.

46. In the remarks, Applicant argues the novelty/rejection of claim 1 on pages 8-9 of the remarks, in substance that:

> "...the negation of the seq_int# does not necessarily indicate that two addresses are not sequential. At most, it indicates that they may not be sequential."
>
> "...the negation of the seq_int# signal indicates that the counting scheme will be interleaved, thus providing an indication of the relation of a subsequent address to a current address but provides no indication of the relation of a current address to a previous one. That is, the seq_int# does not indicate a relation of the current address to a previous one because it is a forward looking signal."

47. These arguments are not found persuasive for the following reasons:

a) Regarding the first argument, the examiner asserts that the seq_int# signal can be one of two possible states. It can either denote a sequential state, where addresses are sequential, or in a second state, where this signal denotes non-sequential (or interleaved mode), as discussed in the "background of the invention" section. If the signal is set to indicate sequential mode, then the current address is sequential to a previous address. And, if the signal is set in non-sequential (interleaved) mode, then the current address is not sequential to a previous address. In interleaved mode, the addresses are interleaved with one another, even if at most two cases result in sequential addressing. It should be realized though that when the first address is 001, the next address is 0, the next address is 3, the next address is 2, the next address 5, and so on (Table 1 in column 6). This is clearly non-sequential addressing, and in this situation, the signal will be set in non-sequential mode.

b) Regarding the second argument, this signal deals with both relationships (which is arguably the same relationship). A current address is obtained based on the value of the previous address. Likewise, a subsequent address is obtained based on the value of the current address. Therefore, this signal does provide an indication of the relationship between a current and previous address

in that the signal indicates that the current address will be interleaved with respect to the previous address. Column 6, lines 20-25 also illustrate this relationship.

48. In the remarks, Applicant argues the novelty/rejection of claim 3 on pages 9-10 of the remarks, in substance that:

> "The signals of Oh cited by the examiner are provided as inputs to a counter to properly control burst access. However, they are not generated by a fetch unit."
>
> "...the seq_int# signal of Oh does not definitely indicate that the current address is not sequential to the previous address, as claimed in claim 3."
>
> "...the negation of the seq_int# signal indicates that the counting scheme will be interleaved, thus providing an indication of the relation of a subsequent address to a current address but provides no indication of the relation of a current address to a previous one."

49. These arguments are not found persuasive for the following reasons:

a) Regarding the first argument, the applicant has not defined, in the claims, what a fetch unit constitutes. Therefore, the examiner feels that an appropriate interpretation of such a unit that causes fetching to occur based on the generated signals. More specifically, the signals taught by Oh are inherently generated by some unit. And these signals are used to control fetching of some type. Therefore, the unit generating these signals can be referred to as a fetch unit.

b) Regarding the second argument, the applicant should see the response to the argument for claim 1 above. In addition, applicant does not claim that the signal **definitely** indicates that the current address is not sequential to the previous address. When the addresses are in fact non-sequential, the seq_int# signal will have been set accordingly to indicate that.

c) Regarding the third argument, the applicant should see the response to the argument for claim 1 above.

50. In the remarks, Applicant argues the novelty/rejection of claim 13 on pages 11-12 of the remarks in a manner similar to the arguments presented for claims 1 and 3. Therefore, the examiner's response to applicant's arguments for claims 1 and 3 above also apply to applicant's arguments for claim 13.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to David J. Huisman whose telephone number is (703) 305-7811. The examiner can normally be reached on Monday-Friday (8:00-4:30).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (703) 305-9712. The fax phone number for the organization where this application or proceeding is assigned is (703) 746-7239.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

DJH
David J. Huisman
January 28, 2004

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100